



COPY OF PAPERS  
ORIGINALLY FILED

Atty. Docket No.: 51876.P288

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re the application of:

Kang Bok Lee, et al.

Serial No.: 10/038,923

Assigned Filing Date: December 31, 2001

For: METHOD FOR GENERATING NODES IN MULTIWAY SEARCH TREE  
AND SEARCH METHOD USING THE SAME

PRIORITY DOCUMENT SUBMITTAL

Hon. Commissioner of  
Patents and Trademarks  
Washington, D.C. 20231

Dear Sir:

Submitted herewith is a document upon which Applicant respectfully requests a convention  
priority for the above-captioned application, namely Korean Patent Application No. 2000 87576  
filed December 30, 2000.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: 2/1/02

By: [Signature]

Eric S. Hyman, Reg. No. 30,139

**CERTIFICATE OF MAILING**

I hereby certify that this correspondence is being  
deposited with the United States Postal Service as  
first class mail in an envelope addressed to: Assistant  
Commissioner for Patents, Washington, D.C.  
20231 on 2-13-02

Melissa Stead  
Melissa Stead

2-13-02  
Date

12400 Wilshire Boulevard  
Seventh Floor  
Los Angeles, California 90025  
(310) 207-3800

RECEIVED  
MAR 18 2002  
TC 1700

COPY OF PAPERS  
ORIGINALLY FILED



대한민국 특허청  
KOREAN INDUSTRIAL  
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Industrial  
Property Office.

출원번호 :  
Application Number

특허출원 2000년 제 87576 호

출원년월일 :  
Date of Application

2000년 12월 30일

출원인 :  
Applicant(s)

한국전자통신연구원

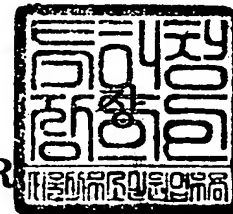
RECEIVED  
MAR 18 2002  
TC 1700



2001      02      22  
년      월      일

특      허      청

COMMISSIONER





920000002923



10111010000000000000

방 식 심 사 란	담	당	심	사	관

【서류명】 특허출원서

【권리구분】 특허

【수신처】 특허청장

【참조번호】 0019

【제출일자】 2000.12.30

【발명의 국문명칭】 다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법

【발명의 영문명칭】 Node Structure of multiway search tree, and search  
Method using it

【출원인】

【명칭】 한국전자통신연구원

【출원인코드】 3-1998-007763-8

【대리인】

【성명】 특허법인 신성 정지원

【대리인코드】 9-2000-000292-3

【포괄위임등록번호】 2000-051975-8

【대리인】

【성명】 특허법인 신성 원석희

【대리인코드】 9-1998-000444-1

【포괄위임등록번호】 2000-051975-8

【대리인】

【성명】 특허법인 신성 박해천

【대리인코드】 9-1998-000223-4

【포괄위임등록번호】 2000-051975-8

【발명자】

【성명의 국문표기】 이강복

【성명의 영문표기】 LEE,Kang Bok

【주민등록번호】 680123-1789725

【우편번호】 361-271

【주소】 충청북도 청주시 흥덕구 복대동 현대2차아파트 211-1403

【국적】 KR

【발명자】

【성명의 국문표기】 이상우

【성명의 영문표기】 LEE,Sang Woo

【주민등록번호】 710131-1036612

【우편번호】 302-120

【주소】 대전광역시 서구 둔산동 향촌아파트 105-1104

【국적】 KR

【발명자】

【성명의 국문표기】 이형섭

【성명의 영문표기】 LEE,Heyung Sub

【주민등록번호】 641023-1408511

【우편번호】 301-132

【주소】 대전광역시 중구 문화2동 극동아파트 103-902

【국적】 KR

【발명자】

【성명의 국문표기】 이형호

【성명의 영문표기】 LEE,Hyeong Ho

【주민등록번호】 550403-1481019

【우편번호】 305-333

【주소】 대전광역시 유성구 어은동 한빛아파트 108-1003

【국적】 KR

【심사청구】 청구

【취지】 특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사

를 청구합니다.

대리인

특허법인 신성 정지원 (인)

대리인

특허법인 신성 원석희 (인)

대리인

특허법인 신성 박해천 (인)

**【수수료】**

【기본출원료】 20 면 29,000 원

【가산출원료】 1 면 1,000 원

【우선권 주장료】 0 건 0 원

【심사청구료】 7 항 333,000 원

【합계】 363,000 원

【감면사유】 정부출연연구기관

【감면후 수수료】 181,500 원

【첨부서류】 1. 요약서·명세서(도면)\_1통

## 【요약서】

### 【요약】

#### 1. 청구범위에 기재된 발명이 속하는 기술분야

본 발명은 다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법 및 그를 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것임.

#### 2. 발명이 해결하고자 하는 기술적 과제

본 발명은, 다중탐색 트리에서 한 노드에서 사용하는 키의 개수에 관계없이 노드에 기록되는 포인터는 1개만 사용하도록 하여 키, 키 포인터 및 노드 포인터가 캐시라인의 크기와 일치되도록 하여, 다중 탐색 트리를 고속화 함은 물론 이로 인해 메인 메모리의 용량을 줄일 수 있도록 한 다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공함에 그 목적이 있음.

#### 3. 발명의 해결방법의 요지

본 발명은, 라우팅 프로토콜을 수행하는 중앙처리수단(control processing unit)과, 패킷 포워딩을 담당하는 포워딩 수단과, 상기 포워딩 수단에서 제공되는 정보에 따라 해당 링크로 데이터를 전송하는 스위칭 수단을 포함하는 시스템에서,

어드레스 룩업을 통해 출력전을 알아내기 위해 이용되는 다중탐색 트리의 노드 구조에 있어서, 상기 다중 탐색 트리를 구성하는 각 노드가, 다수개의 키(K)와, 하나의 키 포인터(Kp)와, 하나의 노드 포인터(Po)로 구성하되, 상기 키 포인터(Kp)는 해당되는 다수개의 키(K)의 개수에 대한 정보 및 다운 노드 주소를 알려주는 포

인터에 대한 정보를 포함하는 다중탐색 트리의 노드 구조를 제공한다.

#### 4. 발명의 중요한 용도

본 발명은 라우터 및 스위치 시스템의 어드레스 탐색 등에 이용됨

#### 【대표도】

도 7

#### 【색인어】

다중탐색, 노드 구조, 키 포인터, 노드 포인터

【발명의 명칭】

다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법 {Node Structure of multiway search tree, and search Method using it}

스

【도면의 간단한 설명】

도 1은 일반적인 기가비트 라우터의 일실시에 구성도.

도 2는 일반적인 바이너리 트리(Binary Tree)의 구조를 보여주는 도면.

도 3은 도 2에 도시된 바이너리 트리(Binary tree)에 대한 경로 압축을 실시한 Patricia Tree의 구조를 보여주는 도면.

도 4는 일반적인 바이너리 트리(Binary Tree)에 대한 레벨 압축의 결과를 보여주는 도면.

도 5는 일반적인 캐시라인에 따른 자료 구조를 보여주는 도면.

도 6은 일반적인 비트리(Btree)의 트리구조를 보여주는 도면.

도 7은 본 발명에 따른 다중탐색 트리를 이용한 탐색을 위한 자료 구조를 보여주는 도면.

도 8은 도 7에 도시된 노드 포인드의 구조를 보여주는 도면.

도 9a 및 도 9b는 본 발명에 따른 다중탐색 트리를 이용한 탐색에 적용되는 탐색 트리를 보여주는 도면.

도 10은 본 발명에 따른 다중탐색 트리를 이용한 탐색에 대한 일실시에 흐름



## 【명세서】

### 【발명의 명칭】

다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법 {Node Structure of multiway search tree. and search Method using it}

### 【도면의 간단한 설명】

도 1은 일반적인 기가비트 라우터의 일실시에 구성도.

도 2는 일반적인 바이너리 트리(Binary Tree)의 구조를 보여주는 도면.

도 3은 도 2에 도시된 바이너리 트리(Binary tree)에 대한 경로 압축을 실시한 Patricia Tree의 구조를 보여주는 도면.

도 4는 일반적인 바이너리 트리(Binary Tree)에 대한 레벨 압축의 결과를 보여주는 도면.

도 5는 일반적인 캐시라인에 따른 자료 구조를 보여주는 도면.

도 6은 일반적인 비트리(Btree)의 트리구조를 보여주는 도면.

도 7은 본 발명에 따른 다중탐색 트리를 이용한 탐색을 위한 자료 구조를 보여주는 도면.

도 8은 도 7에 도시된 노드 포인트의 구조를 보여주는 도면.

도 9a 및 도 9b는 본 발명에 따른 다중탐색 트리를 이용한 탐색에 적용되는 탐색 트리를 보여주는 도면.

도 10은 본 발명에 따른 다중탐색 트리를 이용한 탐색에 대한 일실시에 흐름

도.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

본 발명은 다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법에 관한 것으로, 보다 상세하게는 다중탐색 트리에서 트리의 깊이를 줄여 탐색을 고속화하고자 한 다중탐색 트리를 이용한 탐색 방법 및 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것이다.

일반적인 기가비트용 라우터 시스템의 구조는 도 1에 도시된 바와 같이 인터페이스(interface)는 패킷을 시스템 내부에서 사용하기 위한 형태와 링크(link)에서 사용될 데이터의 형태 사이를 정합하는 기능을 수행하고, 포워딩 엔진(FE : Forwarding Engine)(14)은 수신된 패킷을 분류하고 가공하게 된다. 즉, 상기 포워딩 엔진은 패킷의 목적지가 어느 곳인가를 알아내 어느 링크를 통해 내보내야 할지를 결정한다. 스위칭 페브릭(12)은 상기 포워딩 엔진(14)에서 결정된 정보를 이용하여 실제 링크로 데이터를 보내는 기능을 수행한다.

어드레스 탐색은 어드레스가 사용되는 방법에 따라 exact match(완전정합)방법과 LPM(Longest Prefix Match : 이하, "최대유효 정합방법"이라고함)방법이 있다. 상기 exact match방법은 이더넷 MAC 어드레스와 같이 모든 어드레스의 비트가 일치하는 경우에 사용되고, 상기 최대유효 정합방법은 찾고자 하는 키와 첫 번째

비트부터 가장 길게 일치하는 자료를 찾는 것으로 IP version 4에 사용된다.

상기 최대유효 정합방법은 비교대상과 가장 길게 일치하는 것을 찾음으로 인해 그 노드가 깊어 탐색시간이 길어지는 문제점이 있어, 이를 해결하기 위해 가장 많이 사용되었던 방법이 Patricia tree이다.

도 2는 바이너리 트리(Binary tree)로서, 루트노드를 기준으로 작은 값은 트리의 왼쪽에, 큰 값은 트리의 오른쪽으로 배치하여 트리를 만드는 것이고, 도 3은 도 2의 바이너리 트리(Binary tree)를 변형한 Patricia tree(패트리샤 트리)이다.

여기서, 상기 이진 트리(Binary Tree) 방법은, 루트노드(root node)에서 시작하여 키의 각 비트의 값 0,1에 따라 트리가 리프노드(leaf node)를 만날 때까지 따라가는 형태로 형성되며, 이 형성 과정에서 만난 노드 중 가장 나중에 만난 표시가 되어있는 노드가 찾고자 하는 키와 일치되는 정보를 가지고 있는 노드가 된다.

그러나, 상기의 이진 트리 경우 키의 길이와 같은 횟수 만큼의 노드를 액세스해야 해야하므로 그 경로 길이가 늘어나는 문제점이 있었다. 따라서, 상기 이진 트리의 경로 길이(Path length)를 줄이기 위한 방법이 몇 가지 제안되었다.

이중에, 도 3에 도시된 패트리샤 트리(Patricia Tree)는 이진 트리의 경로 압축(path compression)하는 방법을 통해 경로 길이를 줄이기 위해 분기가 일어나지 않는 경로를 축약하였고, 도 4에 도시된 바와 같이, 이진 트리의 레벨 압축(level compression)하는 방법을 통해 분기가 많은 노드들이 밀집한 부분에서 노드를 축약하고 있다.

상기의 노드 축약 방법과 다른 방법으로 도 5에 도시된 바와 같이, Lampson

은 다중탐색트리(multiway search tree)를 사용하여 최대유효 정합방법이 필요한 인터넷 프로토콜(IP)에 적용하였다.

도 5는 32바이트의 캐쉬 라인을 가지는 다중 탐색 트리를 보여주고 있다. 각 노드는 키(K1, K2...), 포인터(P0, P1, P2, ...) 그리고 키 포인터(PK1, PK2, ...)로 구성되어 있으며, 각 노드는 32바이트로 구성되어 있다.

상기 Lampson이 다중탐색 트리를 사용한 이유는 현대적인 프로세서 구조들이 일반적으로 취하는 계층적 메모리 구조를 최대한 활용하는 것이다. 즉, 심화되는 메모리 속도와 프로세서 속도차를 극복하기 위해 점점 대용량의 캐시가 프로세서에 탑재되고 있다. 이 캐시의 동작은 임의의 메모리 어드레스가 액세스되면 이 메모리 어드레스를 포함하는 캐시라인 전체가 한꺼번에 프로세서의 캐시로 복사되고 캐시 내에 있는 데이터는 프로세서의 속도로 처리된다. 따라서, 상기 다중탐색 트리의 한 노드를 한 캐시라인 크기로 만들면 추가적인 메인 메모리 액세스 없이 분기 수를 대폭 늘릴 수 있어 다중탐색 트리를 사용한다.

상기와 같은 다중 탐색 트리는 디스크와 메인 메모리 사이에서 메모리 계층 구조를 고려하여 고안된 방법이기 때문에 저속의 메모리와 상대적으로 고속인 메인 메모리를 고려하여 한 번에 디스크에서 메인 메모리로 읽어 들인 디스크 블록을 이용한다. 따라서, 추가적인 저속의 디스크 액세스 없이 분기 수를 메인 메모리의 속도로 늘려야 하는데, 이러한 다중 탐색 트리의 대표적인 예가 비트리(Btree)와 그 변형된 트리들이 있다.

도 6은 일반적인 비트리(Btree)의 트리구조를 보여주는 도면이다.

도 6에 도시된 바와 같이, 비트리(Btree)는 바이너리 트리를 개량한 형태로  
서, 자료의 탐색이나 검색에 필요한 알고리즘에 주로 사용된다. 즉, 상기  
비트리(Btree)는 루트노드와 비교를 한 다음 그 크기에 따라 좌우로 나누어지는 것  
은 바이너리 트리와 동일하나, 다음 노드에서는 그 크기에 따라 한번에 여러개를  
비교함으로써 속도가 빠르며 트리의 깊이를 줄일 수 있는 이점이 있다.

그러나, 비트리(Btree)가 사용된 일반적인 경우, 디스크에 저장된 데이터베  
이스를 액세스하기 위해 역시 디스크에 저장된 인덱스 파일을 구성하는 데 사용하  
는데, 이 데이터베이스를 액세스하기 위한 키는 문자열로 구성되어 이 노드에서 비  
교 결과에 따라 연결될 다음 노드를 가리키기 위한 포인터의 길이에 비해 아주 많  
은 메모리를 차지하게 되는 문제점이 있었다.

#### 【발명이 이루고자 하는 기술적 과제】

이에 본 발명은, 상기와 같은 종래의 문제점을 해결하기 위해 제안된 것으  
로, 다중탐색 트리에서 한 노드에서 사용하는 키의 개수에 관계없이 노드에 기록되  
는 포인터는 1개만 사용하도록 하여 키, 키 포인터 및 노드 포인터가 캐시라인의  
크기와 일치되도록 하여, 다중 탐색 트리를 고속화함은 물론 이로인해 메인 메모리  
의 용량을 줄일 수 있도록 한 다중탐색 트리의 노드 구조 및 이를 이용한 탐색방법  
과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매  
체를 제공함에 그 목적이 있다.

## 【발명의 구성】

상기와 같은 목적을 달성하기 위한 본 발명은, 라우팅 프로토콜을 수행하는 중앙처리수단(control processing unit)과, 패킷 포워딩을 담당하는 포워딩 수단과, 상기 포워딩 수단에서 제공되는 정보에 따라 해당 링크로 데이터를 전송하는 스위칭 수단을 포함하는 시스템에서, 어드레스 룩업을 통해 출력전을 알아내기 위해 이용되는 다중탐색 트리의 노드 구조에 있어서, 상기 다중 탐색 트리를 구성하는 각 노드가, 다수개의 키(K)와, 하나의 키 포인터(Kp)와, 하나의 노드 포인터(Po)로 구성하되, 상기 키 포인터(Kp)는 해당되는 다수개의 키(K)의 개수에 대한 정보 및 다운 노드 주소를 알려주는 포인터에 대한 정보를 포함하는 다중탐색 트리의 노드 구조를 제공함을 특징으로 한다.

한편, 본 발명은, 어드레스 룩업을 통해 출력점을 알아내기 위해 다중탐색 트리를 이용하여 탐색하는 방법에 있어서, 입력되는 IP 주소와 키 값을 비교하는 제 1 단계; 상기 제 1 단계의 비교 결과가 일치하면 노드에 포함된 키 포인터를 이용하여 출력점을 탐색하는 제 2 단계; 및 상기 제 1 단계의 비교결과가 일치하지 않으면, 노드 포인트를 검색하여 노드의 종류를 판단하며, 판단된 노드의 종류가 리프노드일 경우, 키 비교 결과를 참조하여 어느 영역에서 일치하는지 파악하여 키 포인터를 구하여 출력점을 탐색하고, 상기 판단된 노드의 종류가 분기노드일 경우, 키가 해당하는 영역을 찾아내어 노드 포인터를 참조하여 다음 노드로 이동하고 상기 제 1 단계부터 반복 수행하여 리프노드에서 일치하는 키나 영역을 탐색하는 제 3 단계를 포함하는 것을 특징으로 한다.

한편, 본 발명은, 어드레스 룩업을 통해 출력점을 알아내기 위해, 프로세서를 구비한 다중탐색 시스템에, 입력되는 IP 주소와 키 값을 비교하는 제 1 기능:

상기 제 1 기능의 비교 결과가 일치하면 노드에 포함된 키 포인터를 이용하여 출력점을 탐색하는 제 2 기능: 및 상기 제 1 단계의 비교결과가 일치하지 않으면, 노드 포인트를 검색하여 노드의 종류를 판단하며, 판단된 노드의 종류가 리프노드일 경우, 키 비교 결과를 참조하여 어느 영역에서 일치하는지 파악하여 키 포인터를 구하여 출력점을 탐색하고, 상기 판단된 노드의 종류가 분기노드일 경우, 키가 해당하는 영역을 찾아내어 노드 포인터를 참조하여 다음 노드로 이동하고 상기 제 1 단계부터 반복 수행하여 리프노드에서 일치하는 키나 영역을 탐색하는 제 3 기능을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공함을 특징으로 한다.

여기서 상술된 목적, 특징들 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이다. 이하 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.

도 7은 본 발명에 따른 다중탐색 트리를 이용한 탐색을 위한 자료 구조를 보여주는 도면이다.

즉, 도 7은 노드 구조를 비트리(Btree)에 적용하여 8-way search tree(한글화?)를 구성한 일실시예 노드 구성으로, 즉, 한 개의 노드는 총 32바이트로, 7개의 키(K1 ~ K7)와 1개의 노드포인터(P<sub>0</sub>) 및 키 포인터(K<sub>p</sub>)로 이루어져 32Byte의 캐시라인과 일치 시킨다. 여기서, 모든 키는 32비트로 구성되어 있고, 포인터는 각각

16비트로 구성되어 있다.

상기 키 포인터(Kp)는 노드의 첫번째 키에 해당하는 포인터를 나타내고 있는 값으로서, 분기노드의 경우 키 값과 일치되었을 때의 해당 포트 번호에 대한 정보를 나타내고, 분기노드일 경우, 최대유효 정합방법을 가능하게 하는 영역정보를 나타내게 된다.

도 8은 도 7에 도시된 노드 포인트의 구조를 보여주는 도면이다.

도 8에 도시된 바와 같이, 노드 포인터(Po)는 키의 개수와 노드 포인터에 대한 정보를 동시에 가지고 있다. 즉, MSB 3비트는 노드에 포함된 키의 개수를 나타내며, LSB 13비트는 트리의 하위레벨의 위치를 알려주는 포인터의 역할을 하게 된다. 이때, 포인터의 값이 모두 "1"이면 해당 노드가 리프노드를 나타낸다. 만약 포인터의 값이 모두 "1"이 아니라면 루트노드(root node)이거나 분기노드(branch node)로서 다음 노드를 찾아가는 포인터로 사용한다.

도 9a 및 도 9b는 본 발명에 따른 다중탐색 트리를 이용한 탐색에 적용되는 탐색 트리를 보여주는 도면이다.

여기서, 도 9a는 탐색에 적용되는 8-way search tree(?)의 예를 보여주고 있는데, 1\*, 1001\* 및 10100\*과 같이 3개의 프리픽스(prefix)로 구성되어 있다고 가정하면, 상기 3개의 프리픽스(prefix)를 기준으로, 해당하는 범위의 키를 6개 생성한다. 여기서, 생성된 6개의 키는 1\* 으로부터 100000과 111111, 1001\*로부터 100100, 100111, 그리고 10100\* 로부터 101000, 101111이다.

도 9b는 6개의 기준으로 생성된 트리의 구조로서, 최대유효



정합방법(Longest prefix matching)을 수행하므로, 각 영역별로 포트 번호가 할당되어 있다. 따라서, 루트 노드에서부터 리프 노드까지 검색하여 일치하는 키를 찾으면 포트에 대한 정보를 구할 수 있으며, 리프 노드에서 일치하는 키가 없으면 해당하는 범위를 통하여 포트에 대한 정보를 찾을 수 있다.

예를 들어, IP 주소가 110000인 패킷이 도착하면 생성된 트리에서 101111과 111111사이의 6번째 키 포인터에 해당되므로 키 포인터 =  $K_p + \text{키 개수} + 6$ 의 메모리 위치의 식에서 해당하는 IP의 포트번호를 알 수 있다. 반면에, 입력된 IP 주소가 키와 일치되면 키 포인터는  $K_p$ 와 위치정보(X)에서 바로 구할 수 있다.

도 10은 본 발명에 따른 다중탐색 트리를 이용한 탐색에 대한 일실시예 흐름도이다.

도 10에 도시된 바와 같이, 검색을 통하여 8-way 노드를 읽은 다음 7개의 키 값을 검색할 IP 주소와 비교한다(101, 102).

상기 과정(102)에서 판단한 결과, IP주소와 키 값이 일치할 경우에는 검색 결과에 따른 키 포인터(Key pointer)를 이용하여 목표 주소지로 찾아가고(103), IP 주소와 키 값이 일치하지 않을 경우에는 노드 포인터의 값을 읽어(104), 이 노드 포인터가 분기 노드인지 리프 노드인지를 판단한다(105).

상기 과정(105)에서 판단한 결과, 노드 포인터가 리프노드일 경우에는 키 비교결과를 참조하여 키의 범위(영역)를 알아내고(106), 키 포인터(key pointer)를 계산하여 목표 주소지로 찾아가는(107).

한편, 상기 과정(105)에서 판단한 결과, 노드 포인터가 분기 노드일 경우에

는 키가 해당하는 영역을 탐색하고(108). 다음 노드로 이동하기 위해 해당하는 포인터를 이용해서 다음 노드 포인터를 계산하고 다음 노드로 이동한 후(109). 키 값을 읽고 키 값들을 비교하는 과정(102)으로 진행하여 최종으로 리프 노드에서 일치하는 키 값이 나오거나 리프 노드에 도달할 때까지 탐색을 반복 수행한다.

상술한 바와 같은 본 발명의 방법은 프로그램으로 구현되어 컴퓨터로 읽을 수 있는 기록매체(씨디롬, 램, 롬, 플로피 디스크, 하드 디스크, 광자기 디스크 등)에 저장될 수 있다.

이상에서 설명한 본 발명은 진술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위내에서 여러가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

#### 【발명의 효과】

상기와 같은 본 발명은, 다중 탐색 트리에서 한 노드에서 사용하는 포인터의 수를 한개만 사용하여 한 개의 캐시라인에 기록할 수 있는 키의 개수를 증가시킬 수 있고, 이에 따라 네트워크 어드레스 탐색에서 분기수가 늘어나 트리의 깊이를 줄일 수 있어 탐색의 고속화하고 포워딩 엔진의 속도를 높일 수 있는 효과가 있다.

또한, 본 발명은 메모리가 절감되어 메모리 요구량이 작아진 부수 효과에 따라 2차 캐시에 들어가는 비율이 높아지게 되어 추가적인 속도 향상 효과를 얻을 수 있다.

【특허 청구범위】

【청구항 1】

라우팅 프로토콜을 수행하는 중앙처리수단(control processing unit)과, 패킷 포워딩을 담당하는 포워딩 수단과, 상기 포워딩 수단에서 제공되는 정보에 따라 해당 링크로 데이터를 전송하는 스위칭 수단을 포함하는 시스템에서,

어드레스 룩업을 통해 출력전을 알아내기 위해 이용되는 다중탐색 트리의 노드 구조에 있어서, 상기 다중 탐색 트리를 구성하는 각 노드가, 다수개의 키(K)와, 하나의 키 포인터(Kp)와, 하나의 노드 포인터(Po)로 구성하되, 상기 키 포인터(Kp)는 해당되는 다수개의 키(K)의 개수에 대한 정보 및 다운 노드 주소를 알려주는 포인터에 대한 정보를 포함하는 것을 특징으로 하는 다중탐색 트리의 노드 구조.

【청구항 2】

제 1 항에 있어서, 상기 노드 포인터(Po)가 모두 '1'이면 해당 노드는 리프 노드(leaf node)인 것을 특징으로 하는 다중 탐색 트리의 노드 구조.

【청구항 3】

제 1 항 또는 제 2 항에 있어서, 상기 다수개의 키(K)는 32비트, 상기 키 포인터(Kp)는,

16비트, 상기 노드 포인터(Po)는 16비트로 구성된 것을 특징으로 하는 다중 탐색 트리의 노드 구조.

【청구항 4】

제 1 항 또는 제 2 항에 있어서,

상기 키(K)의 개수를 7로 하여 8-웨이 서치 트리(8-way search tree)를 구성하는 것을 특징으로 하는 다중 트리의 노드 구조.

【청구항 5】

제 1 항 또는 제 2 항에 있어서,

리프 노드 구성시 키의 영역별로 키 포인터를 할당하여 최대유효정합(largest prefix matching)을 이루는 것을 특징으로 하는 다중 트리의 노드 구조.

【청구항 6】

어드레스 룩업을 통해 출력점을 알아내기 위해 다중탐색 트리를 이용하여 탐색하는 방법에 있어서,

입력되는 IP 주소와 키 값을 비교하는 제 1 단계;

상기 제 1 단계의 비교 결과가 일치하면 노드에 포함된 키 포인터를 이용하여 출력점을 탐색하는 제 2 단계; 및

상기 제 1 단계의 비교결과가 일치하지 않으면, 노드 포인트를 검색하여 노드의 종류를 판단하며, 판단된 노드의 종류가 리프노드일 경우, 키 비교 결과를 참조하여 어느 영역에서 일치하는지 파악하여 키 포인터를 구하여 출력점을 탐색하

고. 상기 판단된 노드의 종류가 분기노드일 경우, 키가 해당하는 영역을 찾아내어 노드 포인터를 참조하여 다음 노드로 이동하고 상기 제 1 단계부터 반복 수행하여 리프노드에서 일치하는 키나 영역을 탐색하는 제 3 단계

를 포함하는 다중탐색 트리를 이용하여 탐색하는 방법.

#### 【장구항 7】

어드레스 특업을 통해 출력점을 알아내기 위해, 프로세서를 구비한 다중탐색 시스템에.

입력되는 IP 주소와 키 값을 비교하는 제 1 기능:

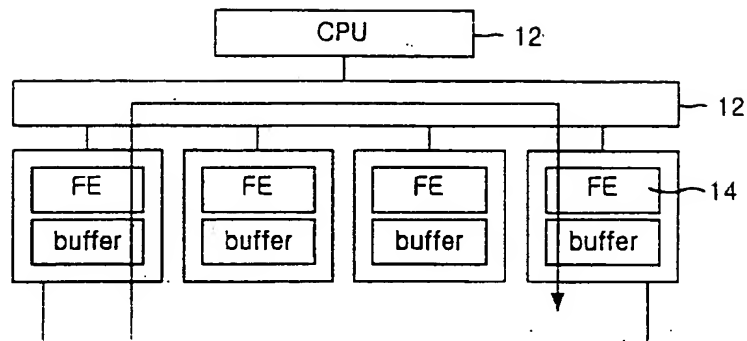
상기 제 1 기능의 비교 결과가 일치하면 노드에 포함된 키 포인터를 이용하여 출력점을 탐색하는 제 2 기능: 및

상기 제 1 단계의 비교결과가 일치하지 않으면, 노드 포인트를 검색하여 노드의 종류를 판단하며, 판단된 노드의 종류가 리프노드일 경우, 키 비교 결과를 참조하여 어느 영역에서 일치하는지 파악하여 키 포인터를 구하여 출력점을 탐색하고, 상기 판단된 노드의 종류가 분기노드일 경우, 키가 해당하는 영역을 찾아내어 노드 포인터를 참조하여 다음 노드로 이동하고 상기 제 1 단계부터 반복 수행하여 리프노드에서 일치하는 키나 영역을 탐색하는 제 3 기능

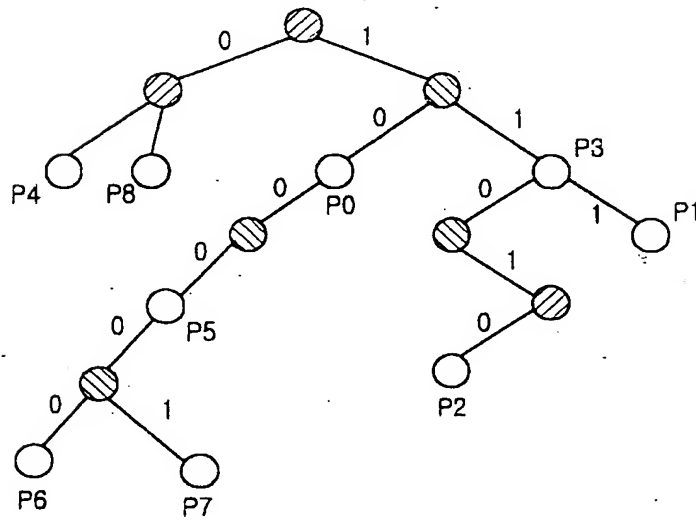
을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

【도 1】

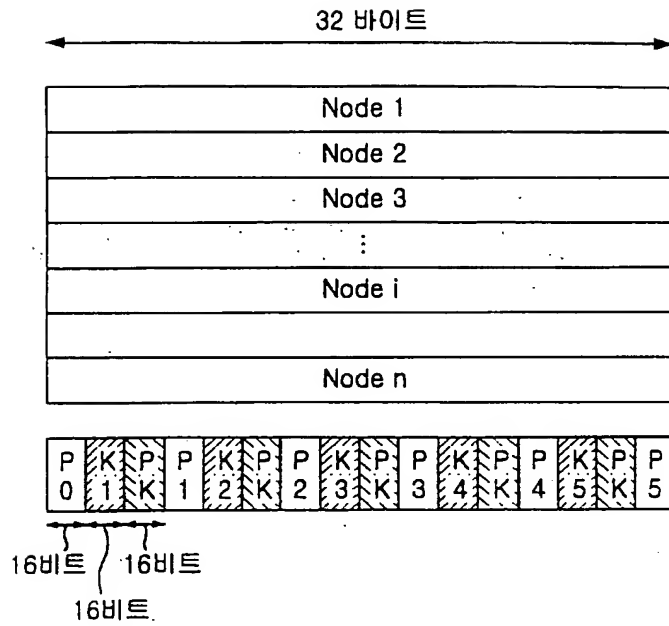


【도 2】

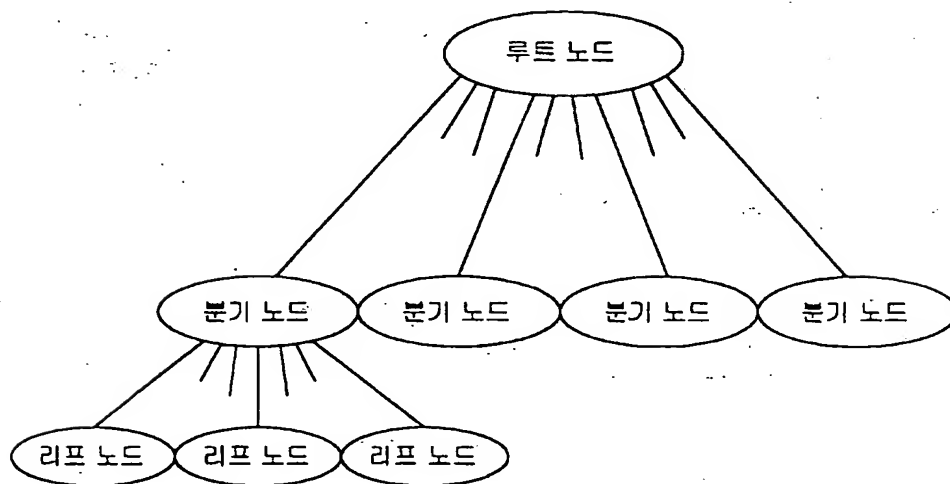




【도 5】

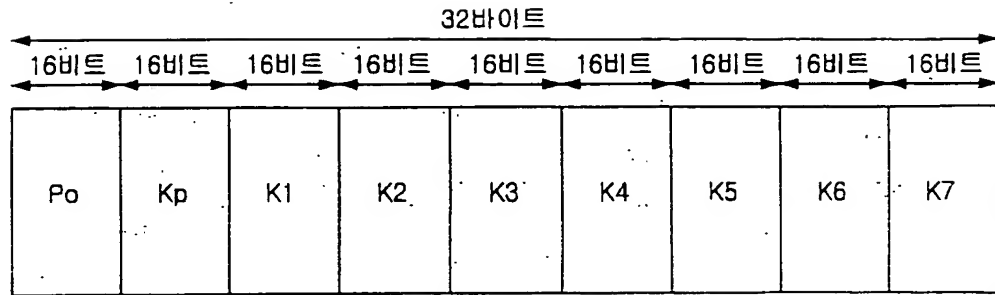


【도 6】

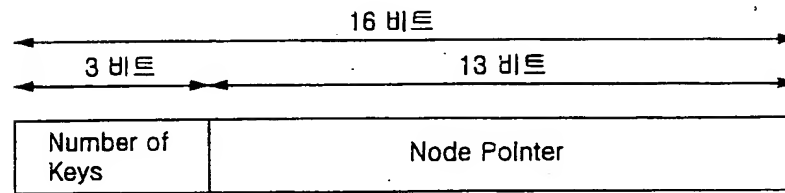




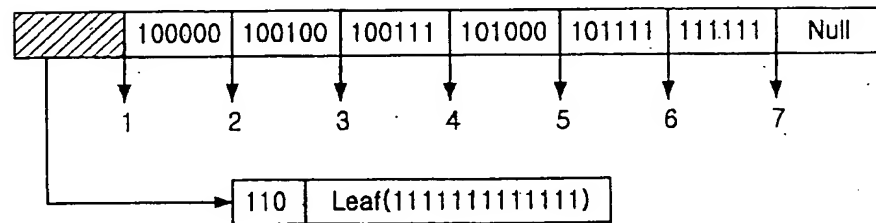
【도 7】



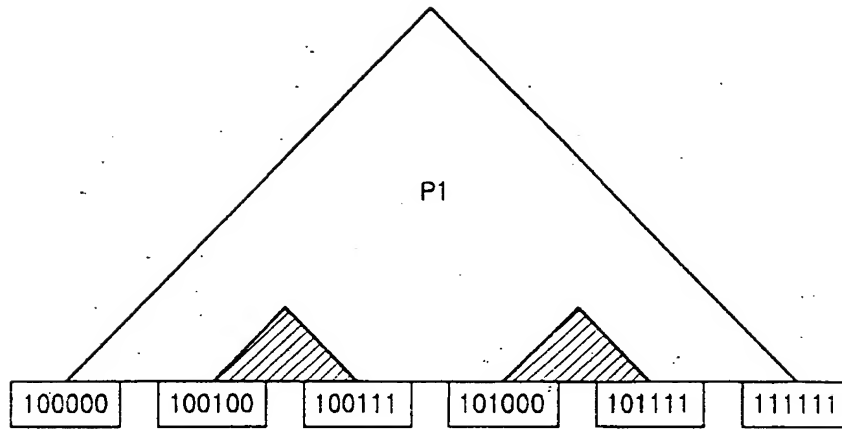
【도 8】



【도 9a】



【도 9b】



【도 10】

